COMPREHENSIVE DESIGN DOCUMENTATON FOR VIDEO PLAYER

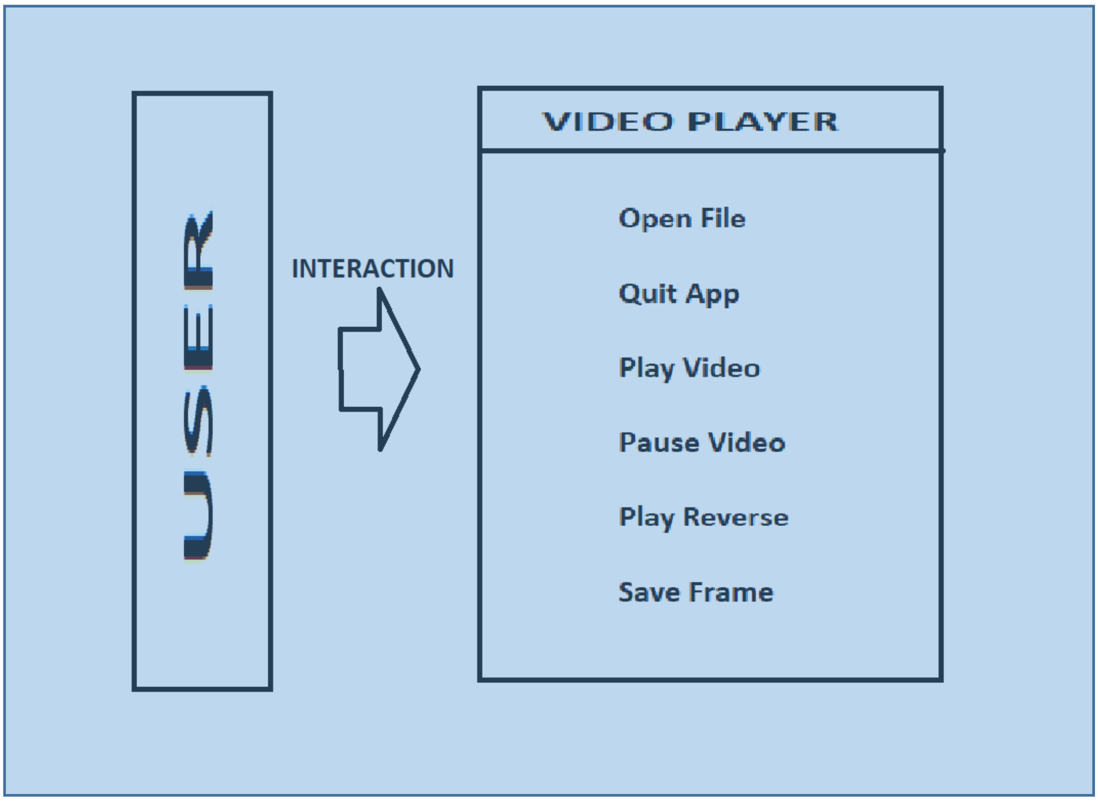
OVERVIEW

The application is a video player developed using PySide6, a Python binding for Qt.
It allows users to open video files (formats supported: .mp4, .avi, .mov).
Key functionalities include playing, pausing, and traversing through the video.
Users can also play the video in reverse.
The application features a slider for navigation through the video.
An overlay, such as a logo or watermark, can be added to each frame of the video.
Additional features include setting playback speed and saving the current frame.

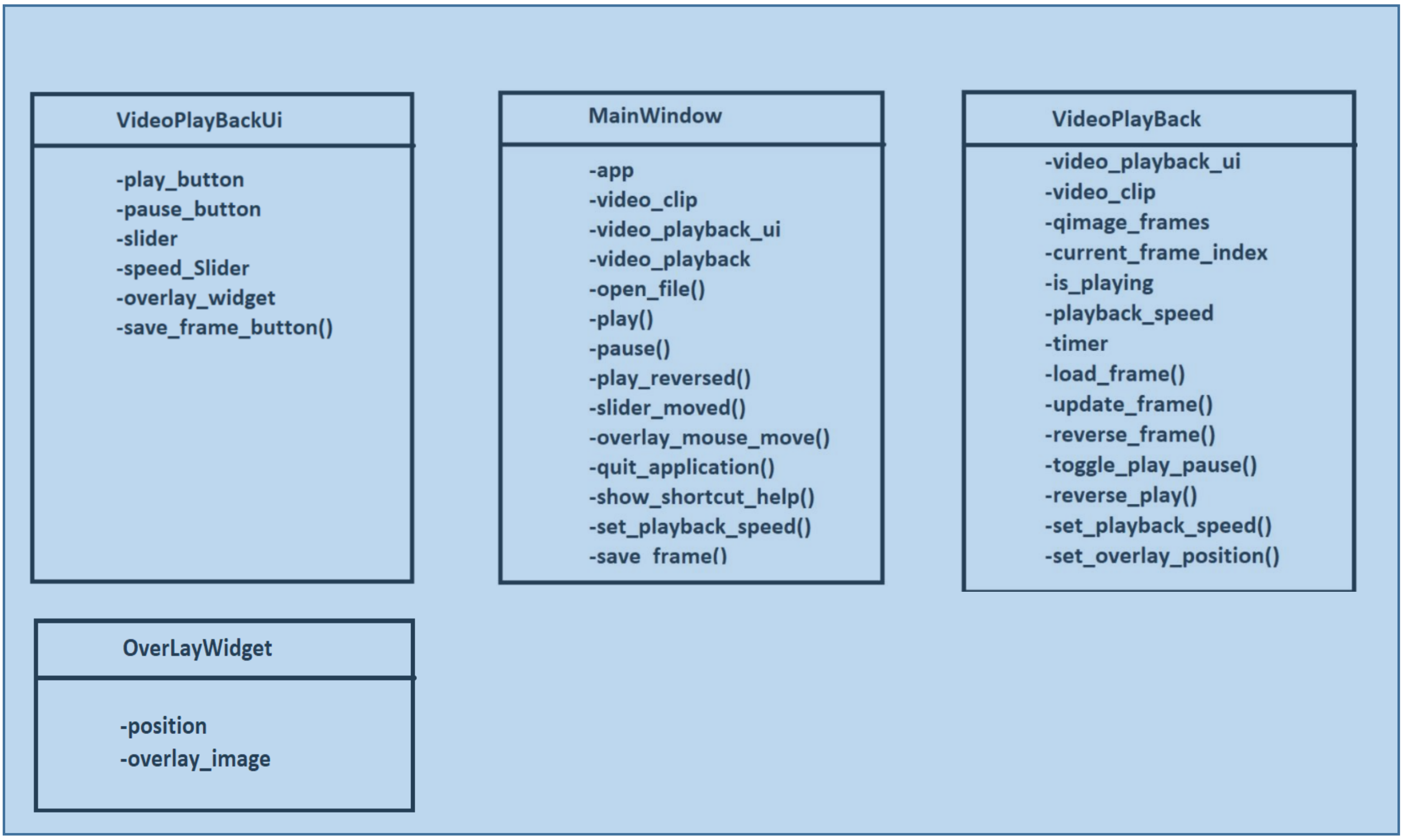
REQUIREMENTS

1. Implement a video player with features like playing, pausing, and navigating through the video.
2. Support playing videos in both forward and reverse directions.
3. Integrate a slider widget for video navigation.
4. Add an overlay on the video.
5. Provide shortcuts for common actions (e.g., open, quit, play reverse).
6. Bonus: Implement a shortcut to save the current frame.
7. Bonus: Enable users to set the playback speed.

USECASE DIAGRAM



CLASS DIAGRAM



DESIGN PATTERN

The **Model-View-Controller (MVC)** design pattern serves as an optimal choice for the video player application, as it inherently fosters maintainability, extensibility, and scalability. By dividing the application into distinct components – Model, View, and Controller – MVC promotes a clear separation of concerns, allowing developers to modify and maintain each component independently. This modular approach not only simplifies debugging and testing but also facilitates the addition of new features or modifications without disrupting existing functionality. Moreover, MVC enables scalability by providing a flexible architecture where components can be scaled individually to accommodate changes in user traffic or data volume. Overall, the MVC design pattern ensures the robustness and flexibility required for developing a high-performing and adaptable video player application.